On Robust Prefix-Tuning for Text Classification

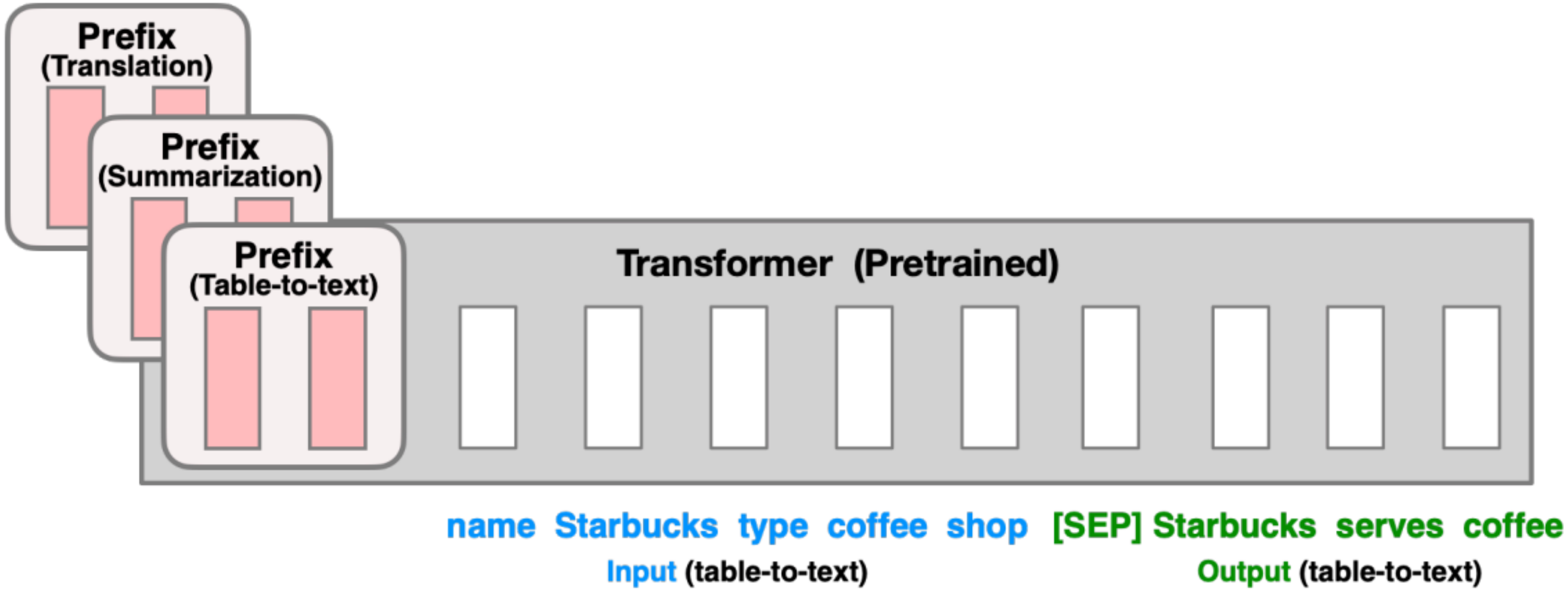
Zonghan Yang @yang_zonghan
Yang Liu @YangLiuNLP
Tsinghua University

Motivation

Prefix-tuning is a parameter-efficient approach to using pretrained LMs.

Prepend **learnable prefixes** to input to steer the pretrained LM.

Optimize only the prefix parameters, with the LM fixed.



Continuous

Modular

Lightweight

Li et al., Prefix-Tuning: Optimizing Continuous Prompts for Generation. TACL'21.

However, the prefix lacks robustness.

Sentiment analysis sentence x

Predict

It ' s just filler .

−

Perturbed sentence $x + \Delta x_i$

It ' s just f ~~i~~ ller .

+

It ' s **just** filler . → **hardly**

+

lifts mates who It ' s just filler .

+

Attacks in NLP:

- Character perturbation
- Synonym replacement
- Sentence rewriting
- Universal adv. triggers

Malicious inputs **fool** the prefix that steers the LM.

Ren et al., Generating Natural Language Adversarial Examples through Probability Weighted Word Saliency. ACL'19.
Li et al., TextBugger: Generating Adversarial Text Against Real-world Applications. NDSS'19.
Wallace et al., Universal Adversarial Triggers for Attacking and Analyzing NLP. EMNLP'19.

Current defense methods are limited for the prefix.

Functional Improvement

Update the word embedding params of the pretrained LM

Adversary Detector

Simultaneously maintain another detector for adversarial inputs

Robustness Verification

Alternate the architecture of the pretrained LM; lack in scalability

Adversarial Training

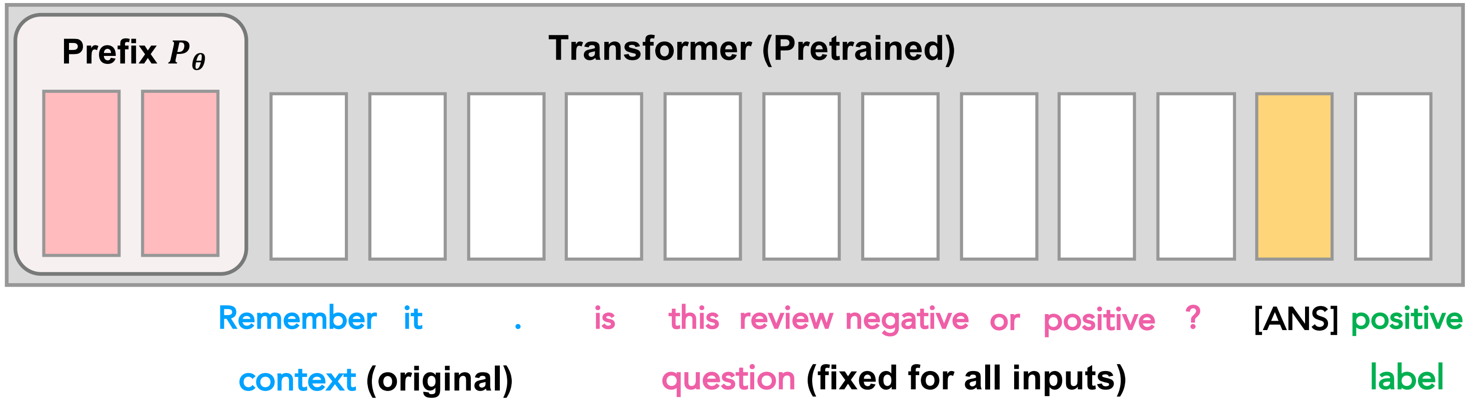
Take much longer time to train (as seen in our experiments)

Jone et al., Robust Encodings: A Framework for Combating Adversarial Typos. ACL'20.
Pruthi et al., Combating Adversarial Misspellings with Robust Word Recognition. ACL'19.
Jia et al., Certified Robustness to Adversarial Word Substitutions. EMNLP'19.
Xu et al., Automatic Perturbation Analysis for Scalable Certified Robustness and Beyond. NeurIPS'20.
Miyato et al., Adversarial Training Methods for Semi-Supervised Text Classification. ICLR'17.
Dong et al., Towards Robustness Against Natural Language Word Substitutions. ICLR'21.

Designing Robust Prefixes

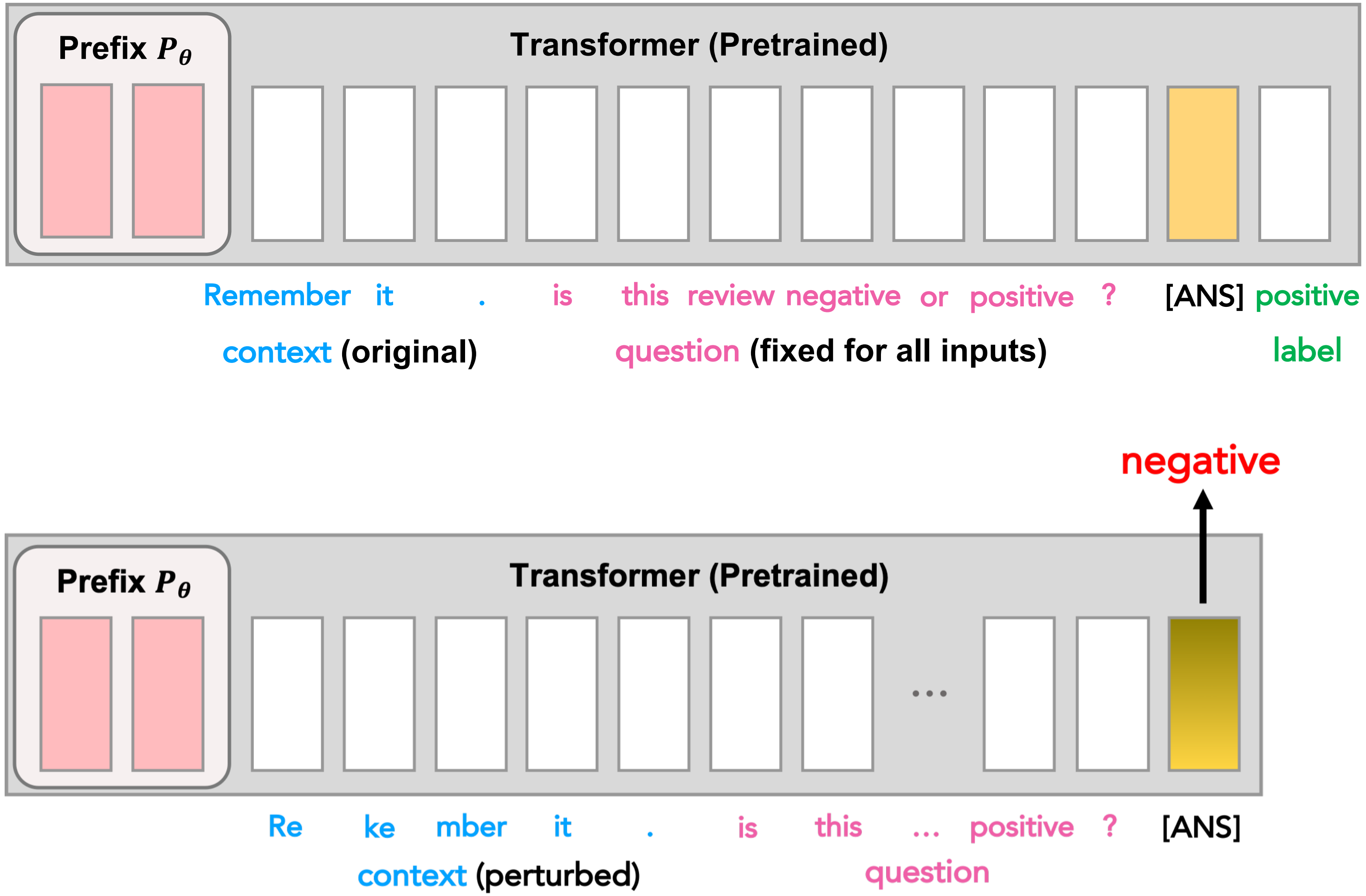
How do we achieve both robustness and parameter efficiency?

Under attack, the LM is mistakenly activated.



LM classifies a sentence by generating the label at the output position (with [ANS] as input).

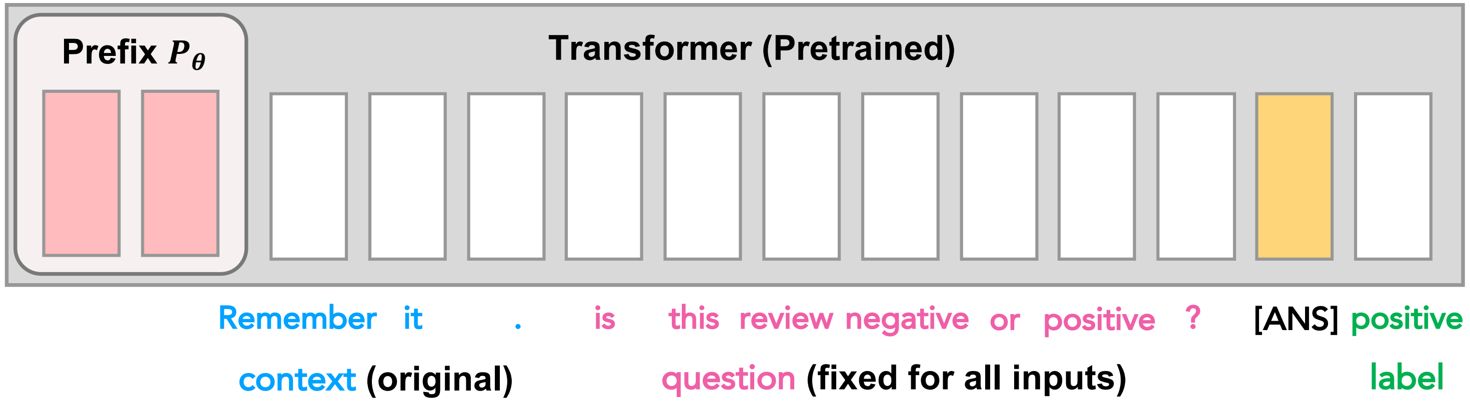
Under attack, the LM is mistakenly activated.



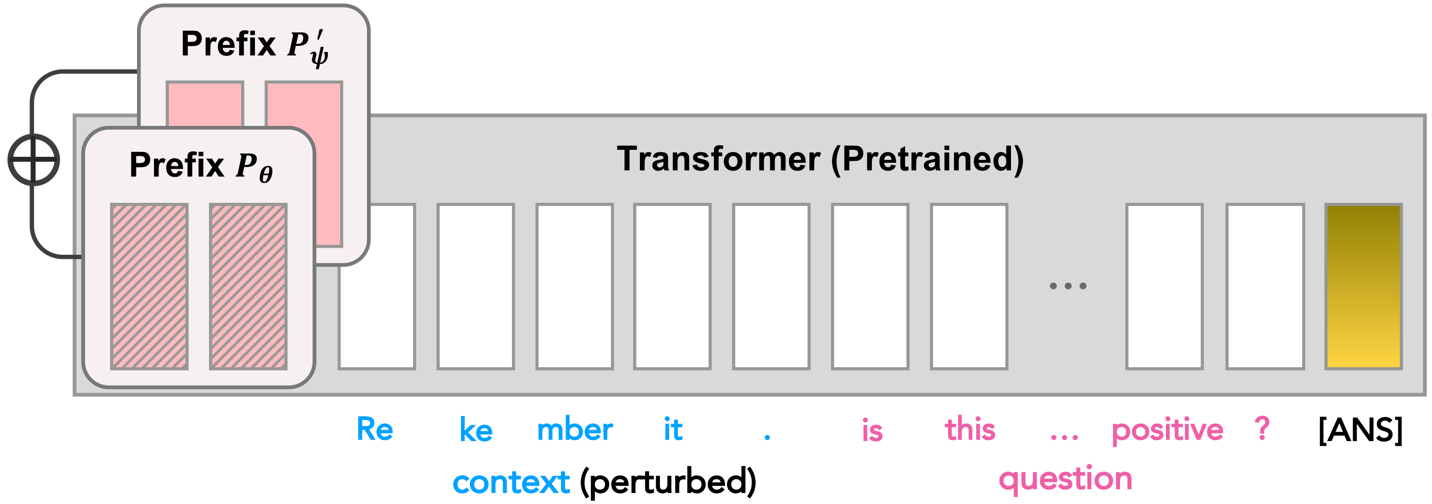
LM classifies a sentence by generating the label at the output position (with [ANS] as input).

While still the [ANS] as input, the LM layerwise activation is erroneous at the output position.

Rectify the activation with an additional prefix!

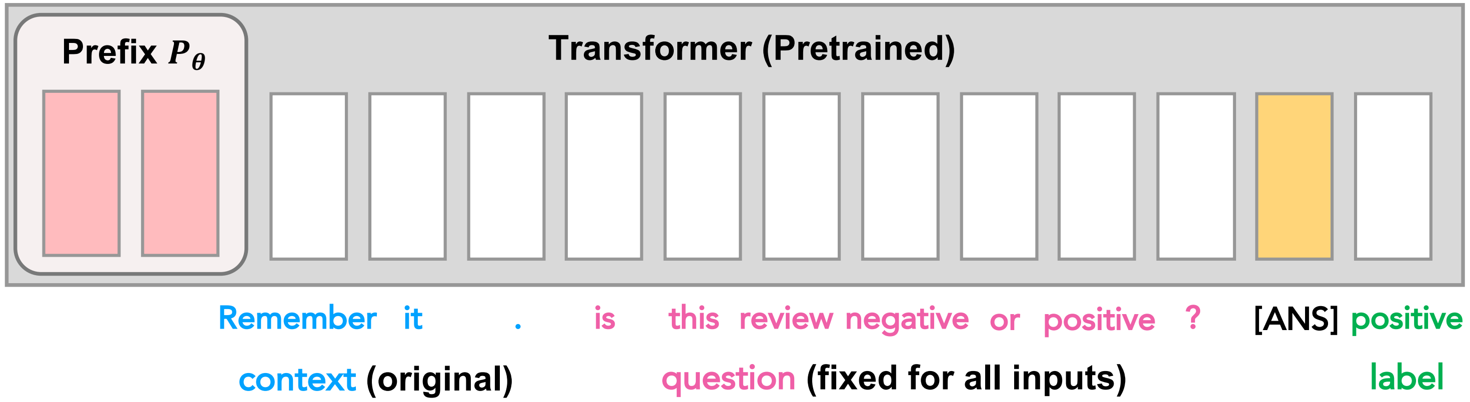


LM classifies a sentence by generating the label at the output position (with [ANS] as input).

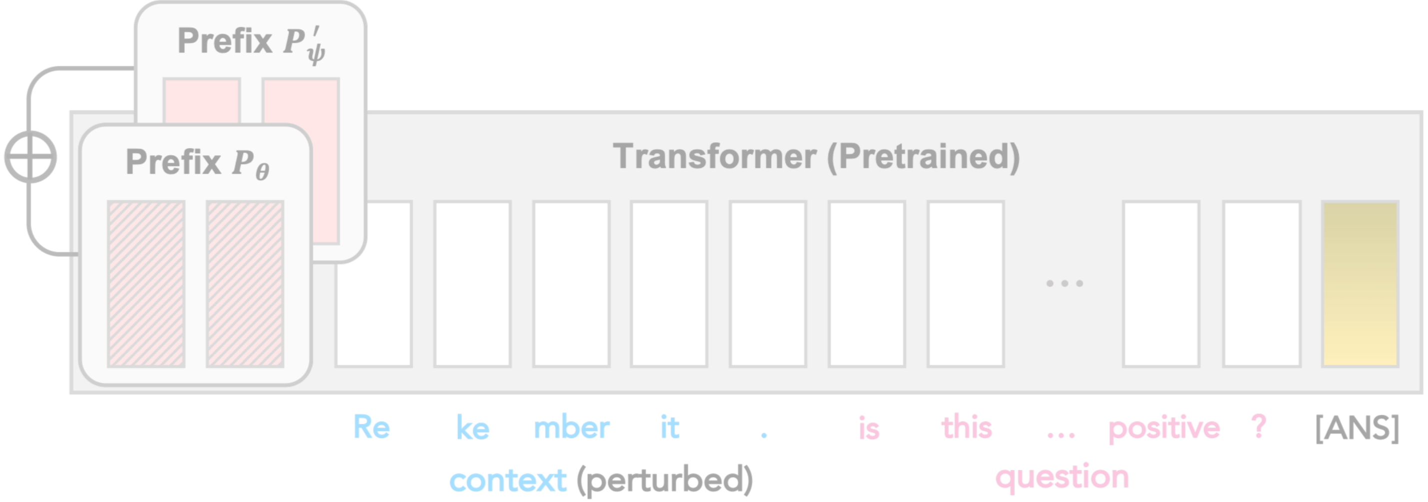


Tune another prefix P'_ψ to rectify the erroneous activation during the inference (w/ P_θ fixed). Still parameter-efficient!

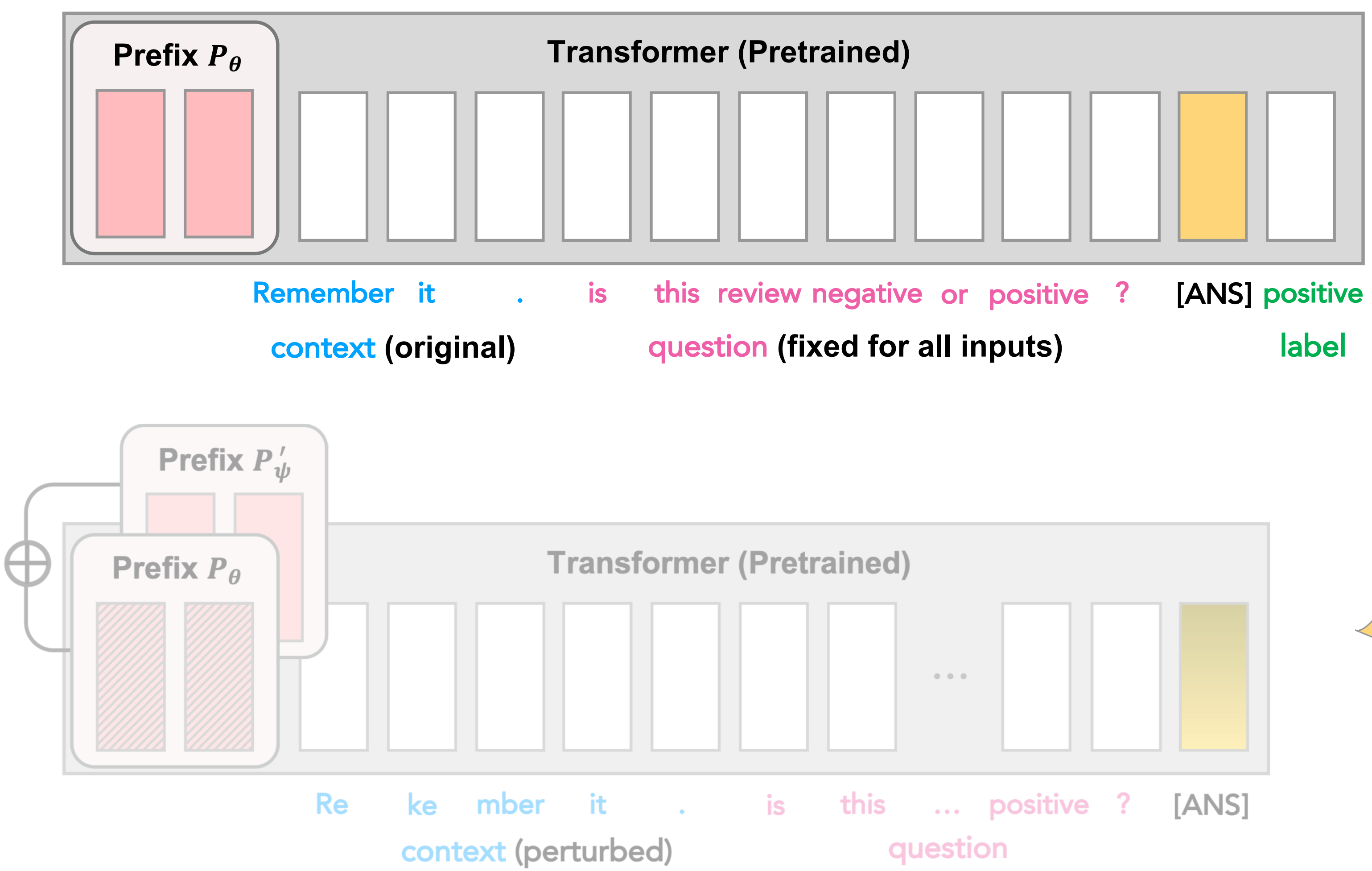
How to tune the additional prefix P'_{ψ} ?



- Get activations of the correctly classified train data under the prefix P_θ.



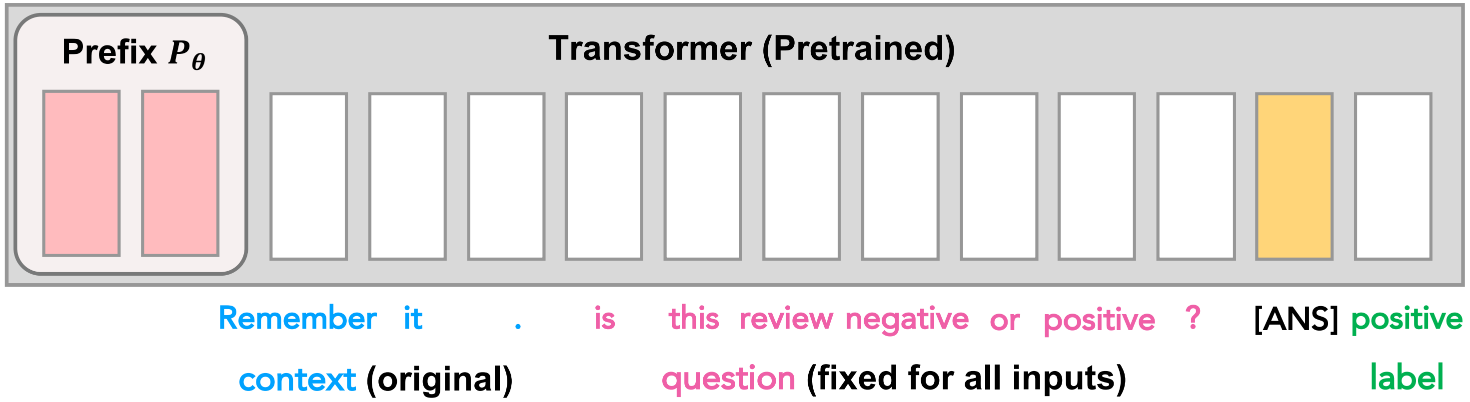
How to tune the additional prefix P'_ψ ?



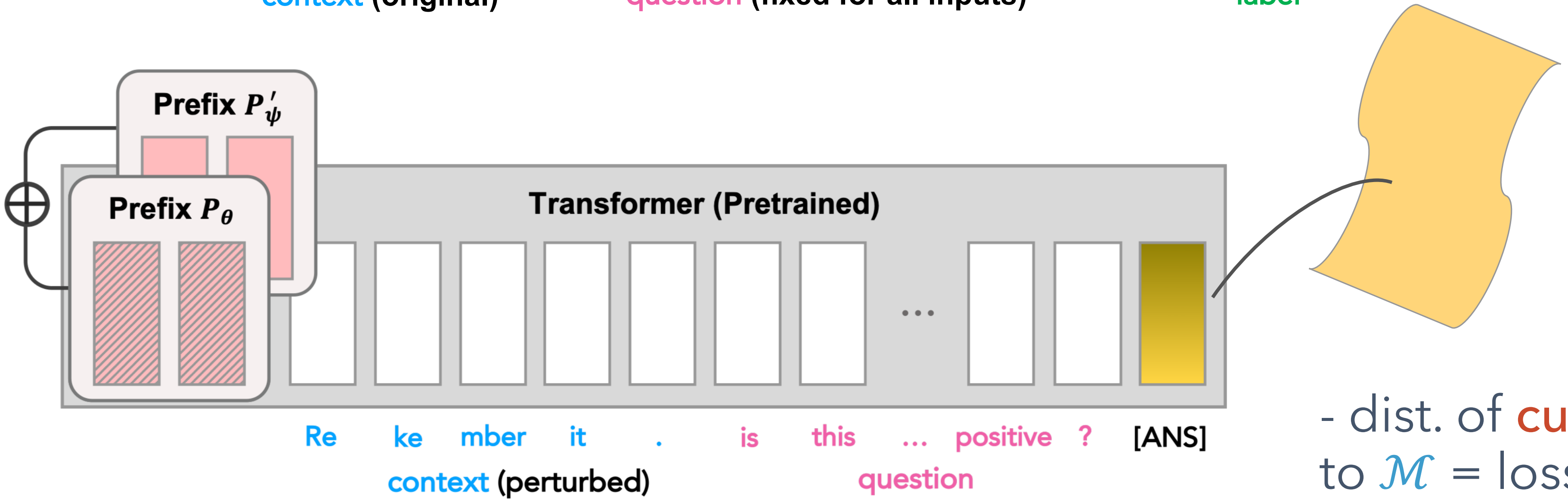
- Get activations of the correctly classified train data under the prefix P_θ.

- Construct the canonical mani-fold $\mathcal{M}$ by PCA.

How to tune the additional prefix P'_ψ ?



- Get activations of the correctly classified train data under the prefix P_θ.



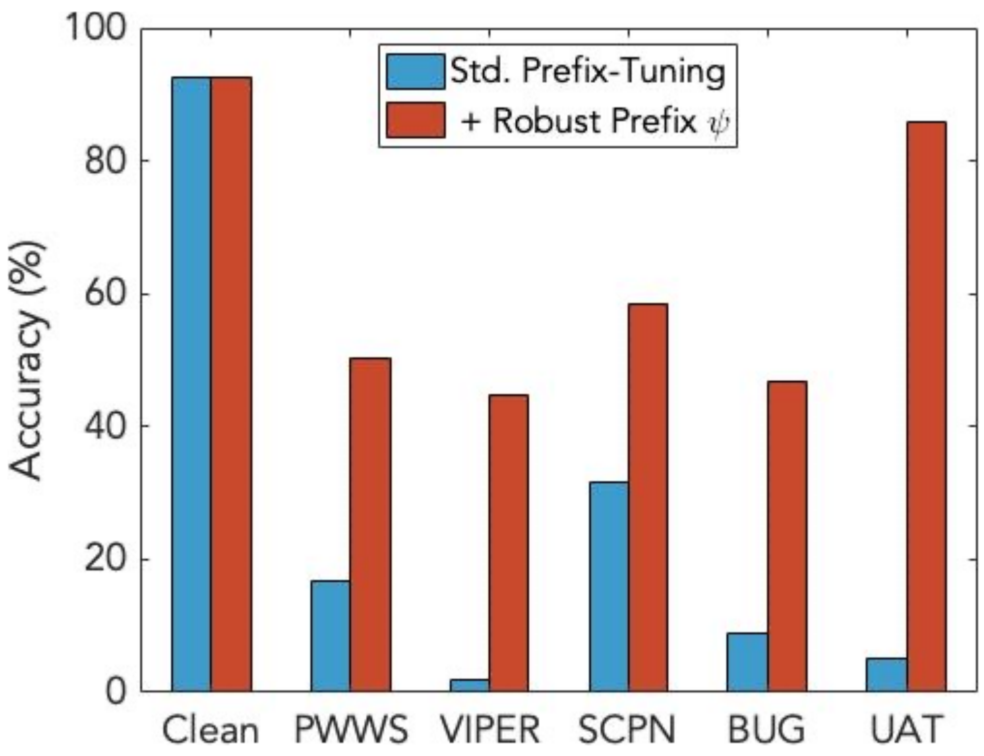
- Construct the canonical manifold $\mathcal{M}$ by PCA.

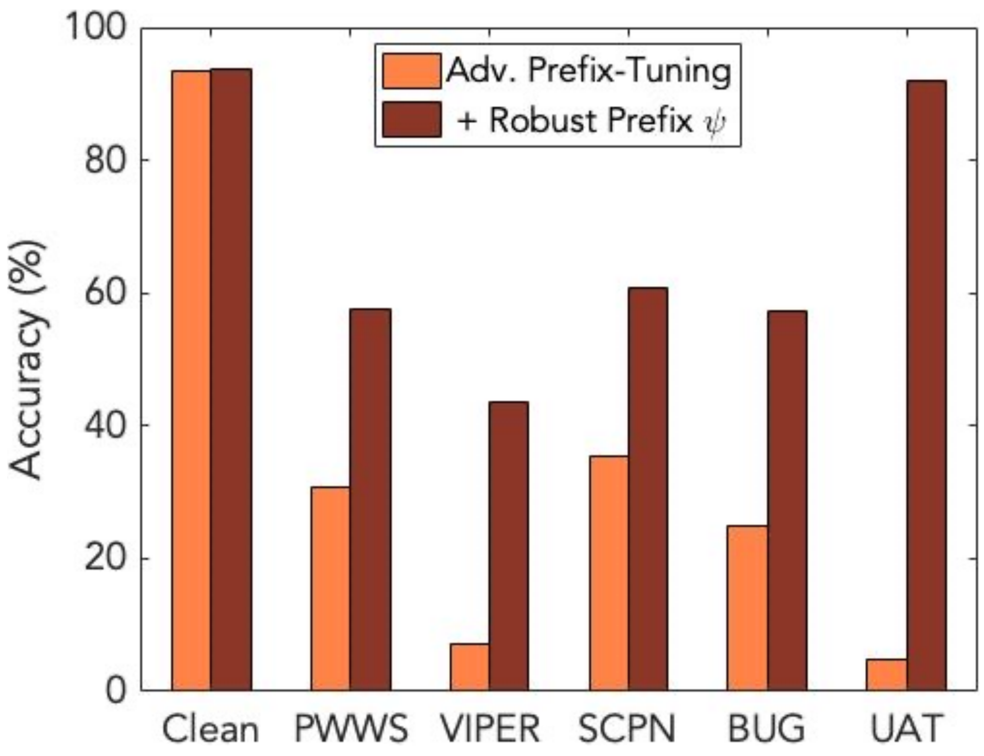
- dist. of current activation to $\mathcal{M}$ = loss for ψ on the fly

Robust Prefix-Tuning: Performance

Clean accuracy retains; robustness seen **significant improvement**

Standard prefix-tuning + **our approach** > **prefix-tuning w/ adv. training**

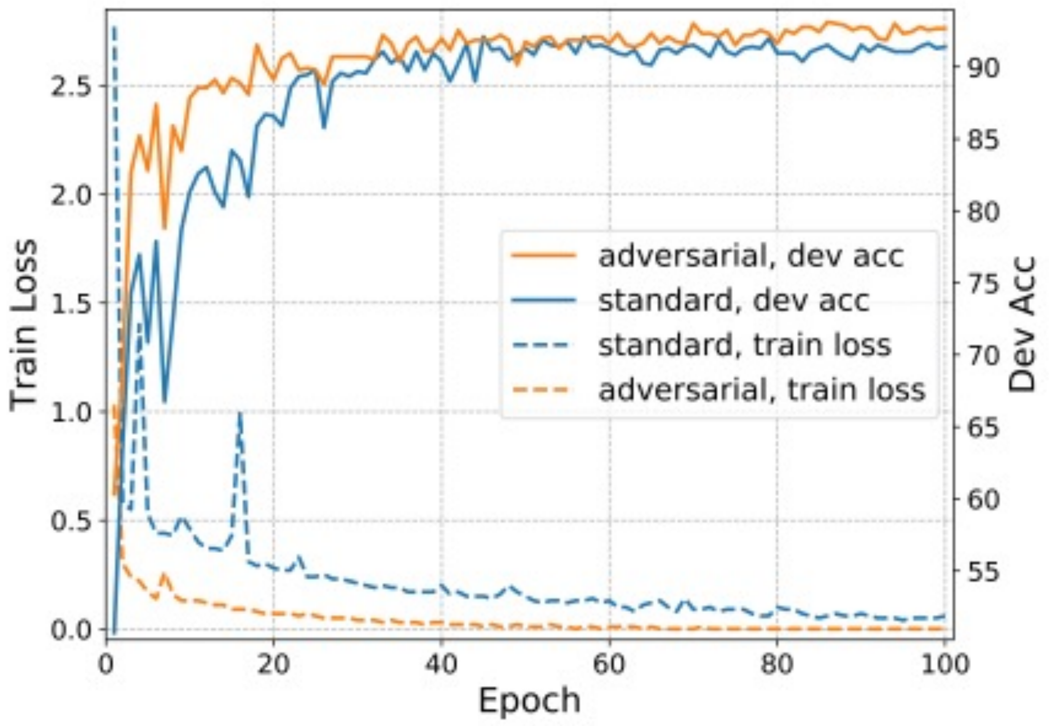


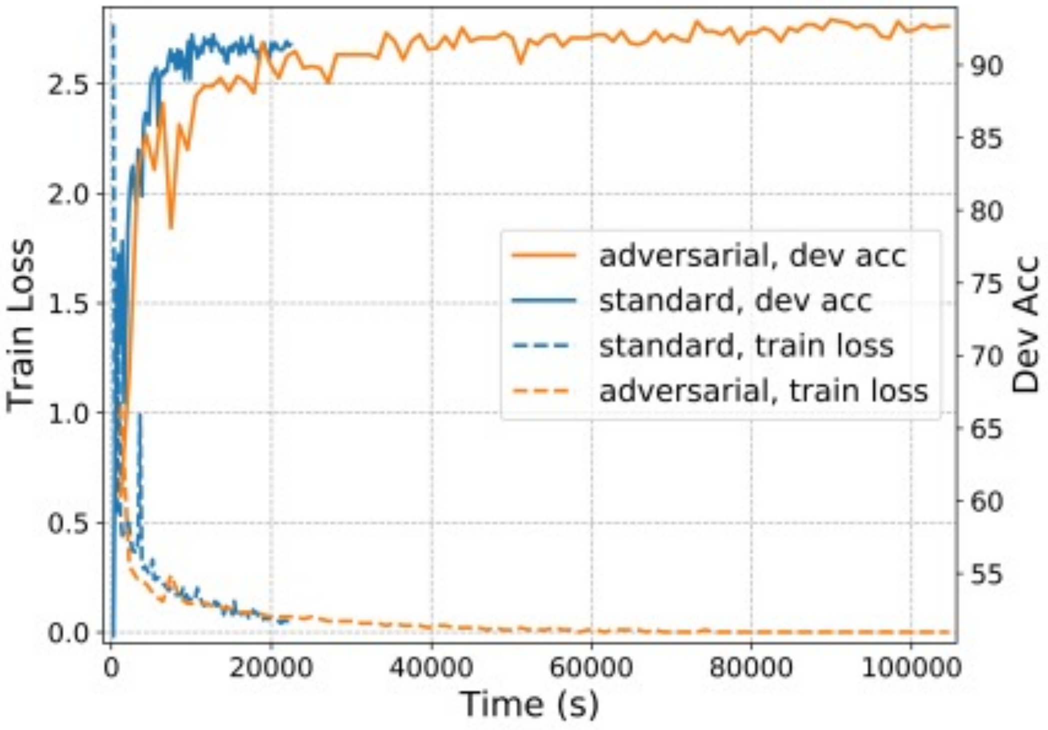


Results from SST-2

Adversarial training is slow for the prefix.

Adv. prefix-tuning improves **epoch**-wise convergence and generalization, but requires much longer **time**!





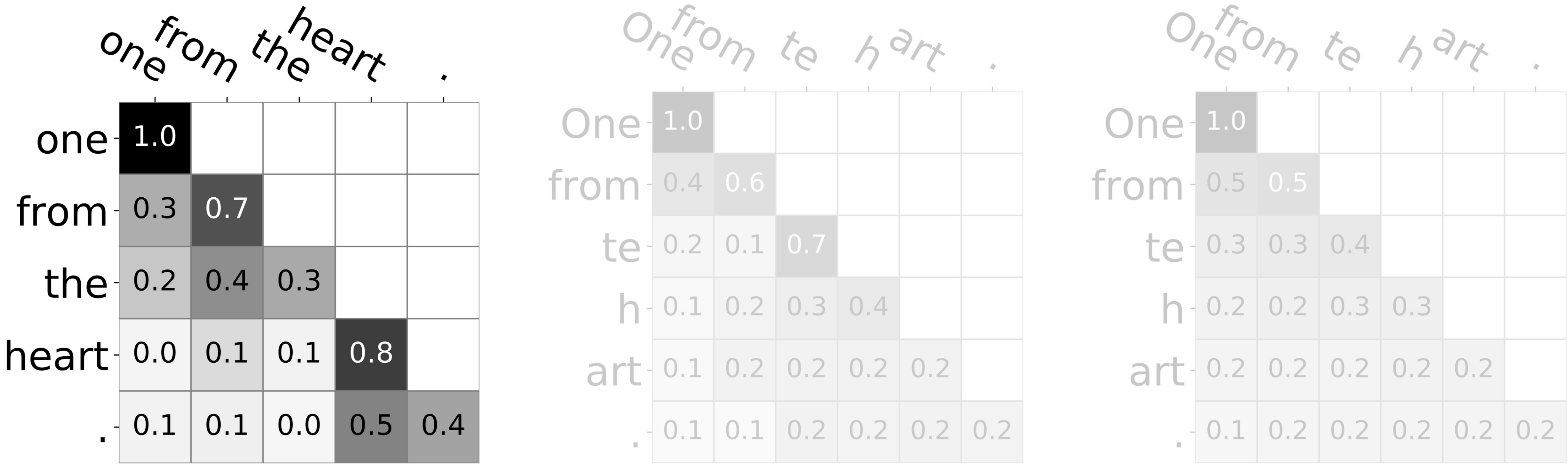
Behavior Analysis

How does robust prefix-tuning steer the LM?

In-sentence Attacks: Averaging the attention

The top-layer attention map of GPT-2 under TextBugger

a. standard prefix-tuning with the original input



a. Predict: positive
std. prefix, ori. input

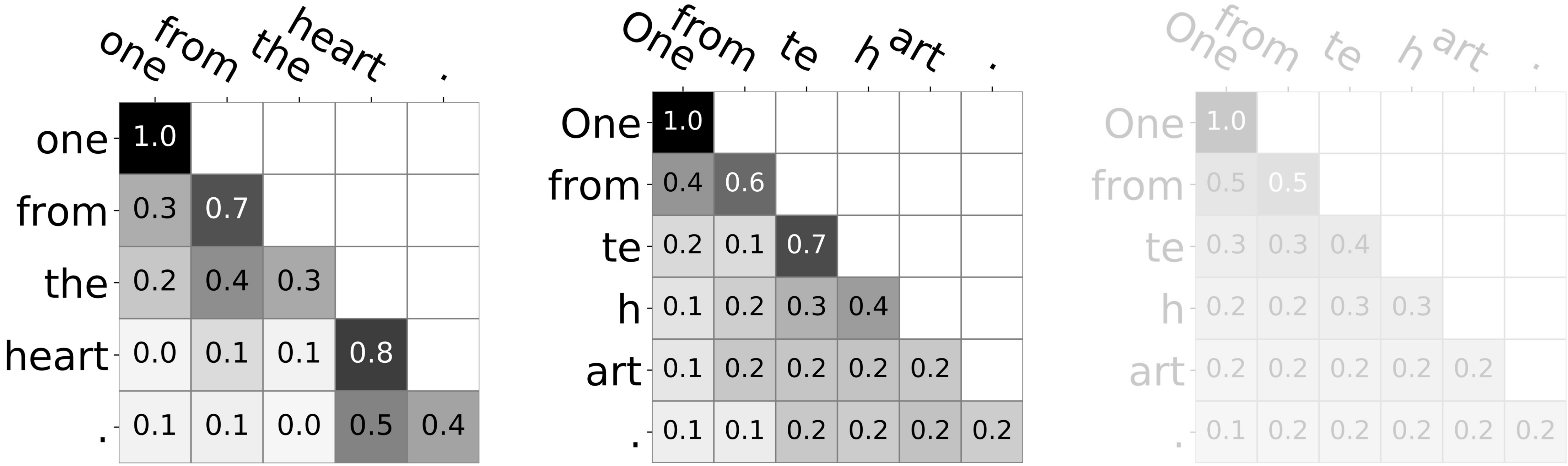
b. Predict: negative
std. prefix, ptb. input

c. Predict: positive
rbs. prefix, ptb. input

In-sentence Attacks: Averaging the attention

The top-layer attention map of GPT-2 under TextBugger

b. standard prefix-tuning with the perturbed input



a. Predict: positive
std. prefix, ori. input

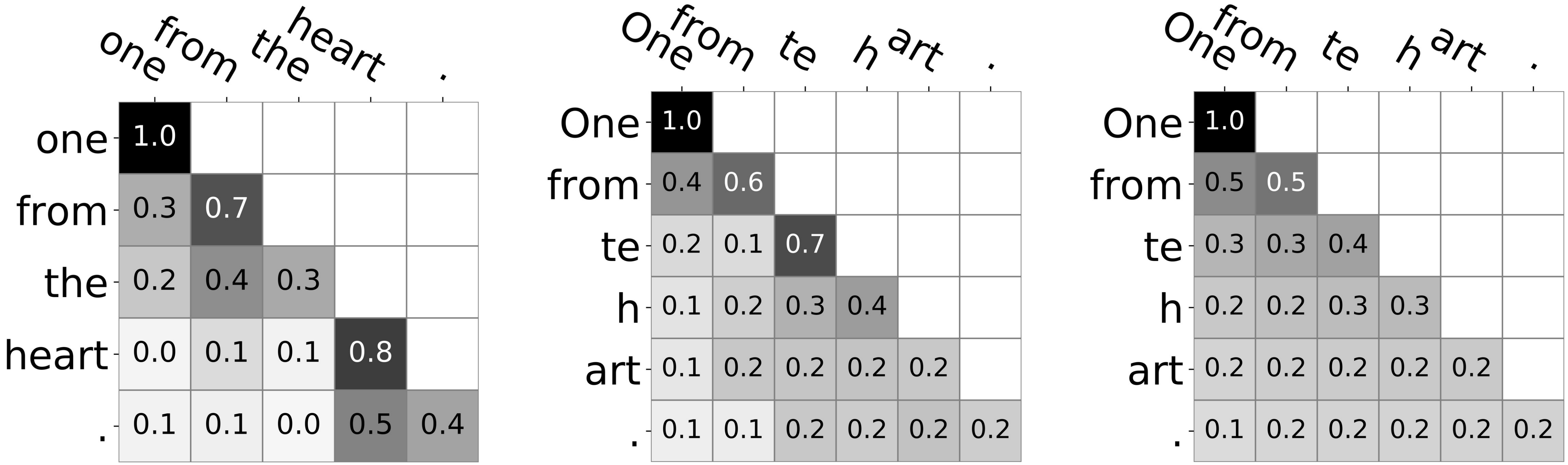
b. Predict: negative
std. prefix, ptb. input

c. Predict: positive
rbs. prefix, ptb. input

In-sentence Attacks: Averaging the attention

The top-layer attention map of GPT-2 under TextBugger

c. robust prefix-tuning with the perturbed input



a. Predict: positive
std. prefix, ori. input

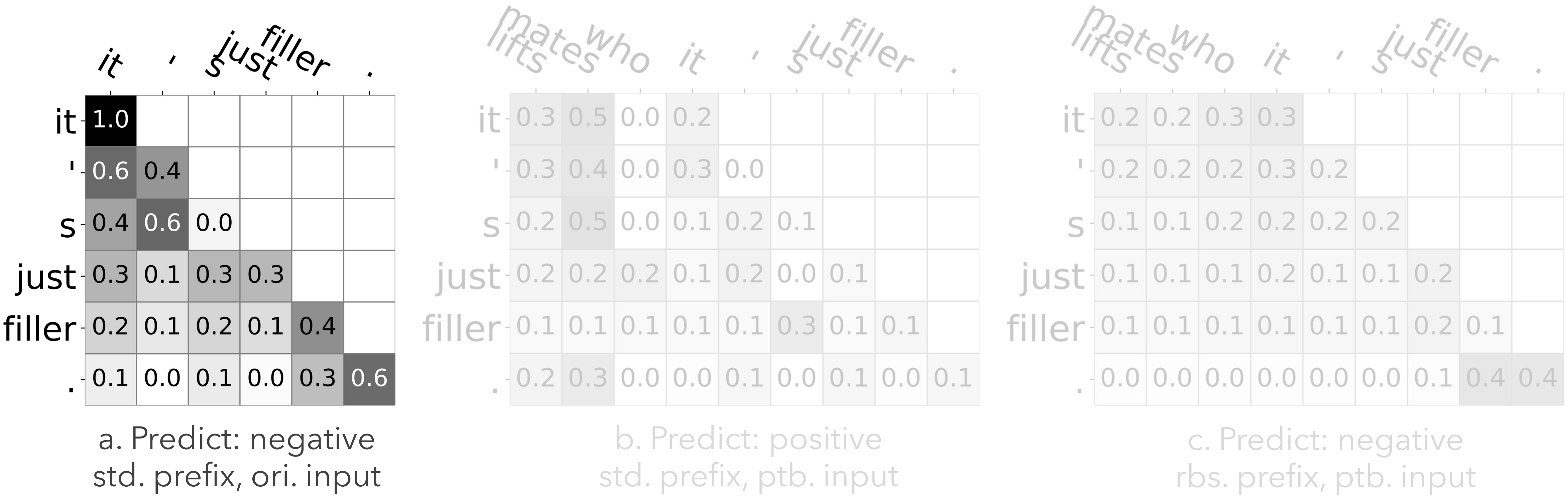
b. Predict: negative
std. prefix, ptb. input

c. Predict: positive
rbs. prefix, ptb. input

Universal Adv. Triggers: Ignoring the distraction

The top-layer token importance map of GPT-2 under UAT

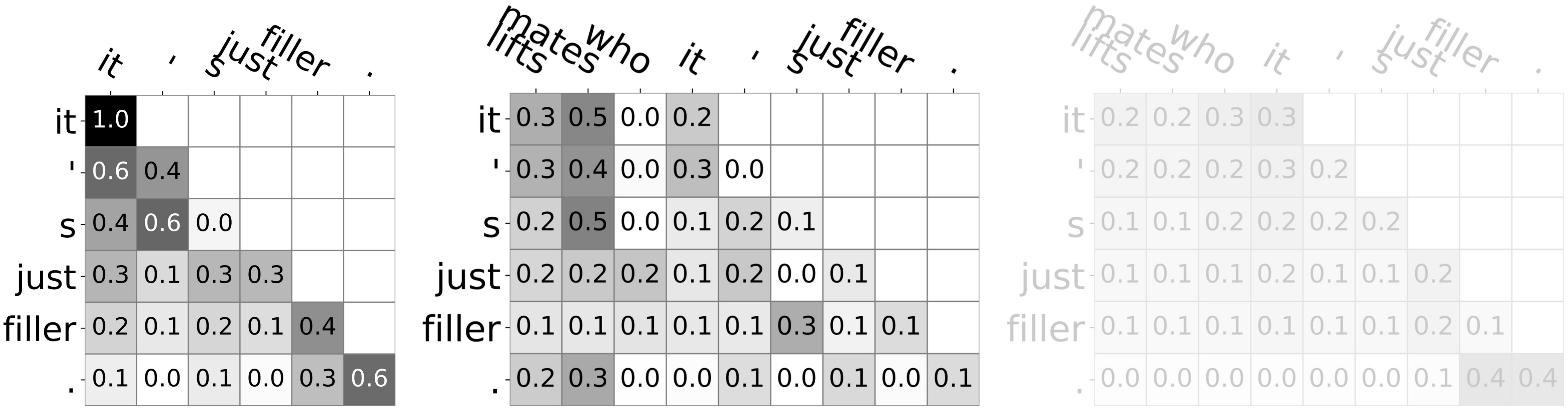
a. standard prefix-tuning with the original input



Universal Adv. Triggers: Ignoring the distraction

The top-layer token importance map of GPT-2 under UAT

b. standard prefix-tuning with the perturbed input – the adversarial triggers **lifts mates who** are prepended to the original test data



a. Predict: negative
std. prefix, ori. input

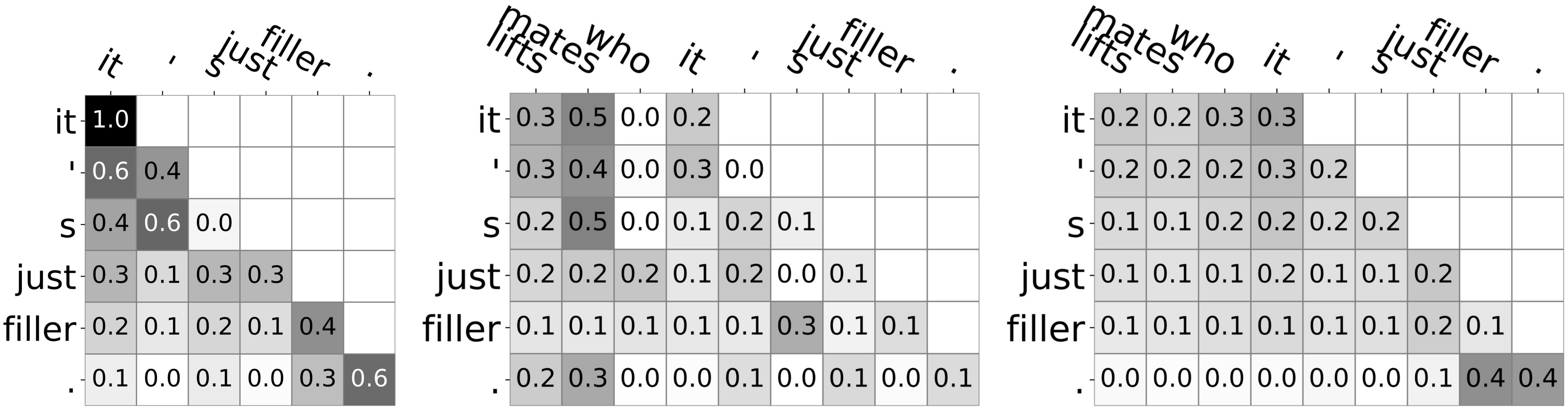
b. Predict: positive
std. prefix, ptb. input

c. Predict: negative
rbs. prefix, ptb. input

Universal Adv. Triggers: Ignoring the distraction

The top-layer token importance map of GPT-2 under UAT

c. robust prefix-tuning with the perturbed input – the adversarial triggers **lifts mates who** are prepended to the original test data



a. Predict: negative
std. prefix, ori. input

b. Predict: positive
std. prefix, ptb. input

c. Predict: negative
rbs. prefix, ptb. input

Robust Prefix-Tuning: The Big Picture

Tune an additional prefix

Lightweight, parameter-efficient

Check out our code! https://github.com/minicheshire/Robust-Prefix-Tuning

Robust Prefix-Tuning: The Big Picture

Tune an additional prefix

Lightweight, parameter-efficient

during inference

Avoid the computational cost of adv. training

Check out our code! https://github.com/minicheshire/Robust-Prefix-Tuning

Robust Prefix-Tuning: The Big Picture

Tune an additional prefix

Lightweight, parameter-efficient

during inference

Avoid the computational cost of adv. training

to steer correct activation

Refine the representation at the output position

Check out our code! https://github.com/minicheshire/Robust-Prefix-Tuning